

Skrypty 3

Celem ćwiczenia jest pogłębienie wiedzy na temat wybranych funkcji systemowych i podstaw programowania w języku bash.

Pliki konfiguracyjne powłoki

Pliki te są zwykle ukryte w naszym katalogu domowym. Zawierają one on wiele poleceń konfiguracyjnych oraz często także komentarze, dzięki którym łatwiej zidentyfikować poszczególne sekcje oraz ich zadania.

W zależności od systemu mogą one mieć różne nazwy. Dla powłok basha będzie to `.bashrc`, `.bash_profile` oraz `.bash_logout`.

Główna różnica w stosunku do plików konfiguracyjnych powłoki polega na tym, że niektóre są odczytywane tylko przez powłoki „logujące” (np. podczas logowania z innego hosta). Są to pliki zawierające w nazwie np. `login`, `profile` lub `zlogin` (w zależności od używanej powłoki).

Kolejna grupa plików konfiguracyjnych to te, które są odczytywane przez „interaktywne” powłoki (np. te podłączone do terminala (lub pseudoterminal w przypadku, emulatora terminala działającego w systemie okienkowym). Są to te, które mają nazwy jak `.bashrc`, `.tcshrc`, `.zshrc` itp.

W przypadku basha sprawa nieco się komplikuje - `.bashrc` jest czytany tylko przez powłokę, która jest zarówno interaktywna, jak i nielogująca, ale w wielu przypadkach „nakazujemy” `.bash_profile`, aby także wykonał `.bashrc` za pomocą:

```
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
```

Kiedy przechodzimy do basha np. pisząc w terminalu `bash` uruchamiany jest skrypt `.bashrc` (uruchamiana jest powłoka interaktywna, nielogująca) Kiedy logujemy się przez `ssh` uruchamiana jest powłoka logująca i interaktywna. W praktyce jednak wykonuje się także `.bashrc`. Z kolei uruchamianie skryptów niewymagających interakcji może się wiązać z uruchomieniem powłoki nieinteraktywnej.

Inne powłoki zachowują się inaczej - np. w przypadku `zsh` plik `.zshrc` jest zawsze odczytywany w przypadku powłoki interaktywnej, niezależnie od tego, czy jest to powłoka logowania, czy nie.

Manual basha wyjaśnia kiedy jest wczytywany każdy plik. Zachowanie to jest ogólnie spójne między maszynami.

FILES

```
/bin/bash
    The bash executable
/etc/profile
    The systemwide initialization file, executed for login shells
/etc/bash.bash_logout
    The systemwide login shell cleanup file, executed when a login shell exits
~/.bash_profile
    The personal initialization file, executed for login shells
~/.bashrc
    The individual per-interactive-shell startup file
~/.bash_logout
    The individual login shell cleanup file, executed when a login shell exits
~/.inputrc
    Individual readline initialization file
```

.profile to po prostu nazwa pliku skryptu logowania oryginalnie używana przez /bin/sh. bash, będący ogólnie kompatybilny wstecz z /bin/sh, odczyta .profile, jeśli taki istnieje.

Aliasy

Aliasy wykorzystujemy w celu utworzenia własnych poleceń, które wykonują inne polecenia. Z reguły celem stosowania aliasów jest zastąpienie złożonych komend prostszymi skrótami.

W celu dodania aliasu należy dokonać stosownej modyfikacji w pliku, zwykle .bashrc

```
alias grepc='grep -rn --colour'
```

Nasz alias nie będzie jednak działał od razu, zgodnie z przytoczonymi powyżej warunkami musi nastąpić zdarzenie, którego konsekwencją jest wykonanie skryptu. Możemy jednak tego dokonać ręcznie.

Pętla while

```
while warunek
do
    ...some code..
done
```

```
while [ $o -lt 3 ]
do
    rm plik$o.txt
    o=`expr $o + 1`
done
```

Pętla until, używana jak pętla while, wykonuje się dopóki warunek jest false.

break – przerywa działanie i opuszcza pętlę.

continue – przerywa bieżącą iterację i rozpoczyna kolejną.

Czy w if może być coś innego niż [/ test?

Tak. Ten przykład sprawdza, czy plik /etc/fstab zawiera słowo "ext4":

```
if grep "ext4" /etc/fstab >/dev/null; then ...
```

Dodanie do wywołania grepa >/dev/null powstrzymuje program grep przed wyświetlaniem na ekran pasujących linii.

Select

Jeżeli już stosujemy w skryptach interaktywność, możemy także użyć instrukcji select. przykład:

```
echo "Wanna see manual for ls command?"
```

```
select i in Yes No
```

```
do
```

```
  case $i in
```

```
    Yes)
```

```
      echo `man ls`
```

```
      break
```

```
    ;;
```

```
    No)
```

```
      break
```

```
    ;;
```

```
    *)
```

```
      echo "Incorrect input"
```

```
    ;;
```

```
  esac
```

```
done
```

Shift

polecenie shift powoduje, że parametry przesuwane są o 1 w lewo. Argument \$1 jest usuwany. Jeżeli nasz skrypt jako ostatni argument przyjmuje nazwę programu do uruchomienia, usunięcie początkowych parametrów shiftem umożliwi uruchomienie programu z jego parametrami przez \$*. Przykład użycia:

```
shift 3
```

Zadanie (opcjonalne):

Zmodyfikuj stosowny plik konfiguracyjny, tak, aby po zalogowaniu się na serwer wyświetlało się imienne powitanie oraz fakt logowania był odnotowywany w ukrytym w katalogu domowym pliku w formie:

nazwa użytkownika - data i czas - logowanie / wylogowanie

Zadanie:

Pobierz i rozpakuj paczkę Paczka.tar. Następnie napisz skrypt, który dla katalogu podanego jako parametr (przeznacz katalog z plikami z paczki) pokaże w konsoli, czy dany plik jest skryptem basha, sh, czy innego języka.

Następnie utwórz w katalogu z plikami paczki podkatalog o dowolnej nazwie i przenieś do niego dwa wybrane pliki. Następnie napisz skrypt, który przeniesie wszystkie pliki o podanym przez użytkownika rozszerzeniu z katalogu paczki i utworzonego wcześniej podkatalogu do folderu podanego jako argument. Rozpatrz przypadek nieistnienia takiego folderu (należy go wtedy stworzyć).

```
#!/bin/bash
```

```
SHEBANG_BASH='#!/bin/bash'
```

```
SHEBANG_SH='#!/bin/sh'
```

```
files=`ls $1` # listuje katalog podany jako parametr  
for file in ${files} #przejdzie po wszystkich plikach  
do
```

```
    case `head -n1 "$1/${file}"` in  
        #head drukuje naglowek pliku - przy  
        #opcji -n drukuje podana liczba linii. Tutaj -n1, czyli pokaze nam pierwsza linie - czyli shebang  
        ${SHEBANG_BASH})  
            echo $file is BashScript  
        ;;
```

```
        ${SHEBANG_SH})  
            echo $file is ShScript  
        ;;  
        *)  
            echo $file is Other  
    esac  
done
```

```
#!/bin/bash
```

```
fileslist=`find sciezka -name ".*$1"`
```

```
for file in $fileslist
```

```
do
```

```
mv ${file} ${2}
```

```
done
```